HAL
archives-ouvertes.fr

Crowd Event Recognition using HOG Tracker

Carolina Garate, Piotr Bilinski, François Bremond

To cite this version:

Carolina Garate, Piotr Bilinski, François Bremond. Crowd Event Recognition using HOG Tracker. Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), Dec 2009, Snowbird, UT, United States. pp.1-6, 10.1109/PETS-WINTER.2009.5399727 . inria-00515197v2

HAL Id: inria-00515197

https://hal.inria.fr/inria-00515197v2

Submitted on 14 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Crowd Event Recognition Using HOG Tracker

Carolina Gárate
Pulsar
INRIA
Sophia Antipolis, France

Piotr Bilinski
Pulsar
INRIA
Sophia Antipolis, France

Francois Bremond
Pulsar
INRIA
Sophia Antipolis, France

Abstract

The recognition in real time of crowd dynamics in public places are becoming essential to avoid crowd related disasters and ensure safety of people. We present in this paper a new approach for Crowd Event Recognition. Our study begins with a novel tracking method, based on HOG descriptors, to finally use pre-defined models (i.e. crowd scenarios) to recognize crowd events. We define these scenarios using statistics analysis from the data sets used in the experimentation. The approach is characterized by combining a local analysis with a global analysis for crowd behavior recognition. The local analysis is enabled by a robust tracking method, and global analysis is done by a scenario modeling stage.

1. Introduction

Just two decades ago, computer vision community had started to focus on the study of crowds in public areas or during public events [1]. This study is motivated by the increasing need for public safety and the high level of degeneration risk especially when a large number of people (crowd) is involved.

In the research field related to crowd analytics we can find different sub-topics like crowd density estimation, crowd tracking, face detection and recognition in crowds, crowd behavior analysis, among others. We are interested in crowd behavior analysis, which is a newest area in the research community. Our goal is to automatically recognize crowd abnormal events in video sequences. In general, the usual process for activity analysis in a video sequence is composed of the following three stages [4]: (1) detection, (2) tracking and (3) event recognition. This process can be applied to crowds as well as individuals.

We propose a new approach for crowd event recognition. The paper considers the second and the third stage of the process mentioned above, to improve the recognition stage. For this purpose in the tracking stage we compute, for every detected object in the first stage (detection), feature points (i.e. corner points) using FAST approach [2]. Then for each computed feature point we build a descriptor based on Histogram of Oriented Gradients (HOG) [3], to finally track these feature points through its respective descriptors. Finally, in the last stage (event recognition) we statistically analyze the vectors formed by the tracking of the feature points, to recognize a pre-defined event.

2. Previous Work

Nowadays, there are many research works related to crowd. The existent approaches in this field can be classified in two categories [5]. One of them is related to crowd event detection, and the other, to crowd density estimation. Some approaches for the second category are based on counting, either: faces, heads or persons [10, 11] but their performance is low when there are occlusions. There are also approaches based on texture and motion area ratio [6, 7, 8, 9], which are really useful for analysis for crowd surveillance. However, neither of them work for event recognition because they cannot detect abnormal situations.

Most of the methods in the first category aim at detecting abnormal events in crowd flows using motion patterns. Motion patterns correspond either to normal behaviors (frequent patterns) or abnormal behaviors (unusual patterns) [12, 13]. For example, *Ihaddadene et al.*[12] approach detects abnormal motion variations using motion heat maps and optical flow. They compute points of interest (POI) in the regions of interest given by the maps. The variations of motion are estimated to highlight potential abnormal events using a pre-defined threshold. The approach does not need a huge amount of data to enable learning pattern frecuency but it is necessary to carefully define, in advance, an appropriate threshold and the regions of interest for every scenario. *Mehran et al.* [13] propose to use social force model for the detection of abnormal behaviors in crowds. The method consists in matching a grid of particles with the frame and moving them along the underlying flow field. Then the social force is computed between moving particles to extract interaction forces, to finally determine the on going behavior of the crowd through the change of interaction forces in time. The resultant vector field is denoted

as force flow, and is used to model the normal behaviors. The method captures the dynamics of crowd behavior without using object tracking or segmentation, nevertheless the obtained false positives could be problematic.

The tracking stage is another topic for the vision community. In the literature we can find several approaches for object tracking trying to solve the occlusion problem. Nevertheless, handling the occlusion for tracking people in crowd is often a harder problem to solve than for tracking individual. Most of the methods for tracking individuals with occlusion may not be so scalable to crowds. One scalable method is KLT [15], which tracks feature points allowing multiple object tracking. *Kaniche et al.* [16] propose a HOG tracker for gesture recognition, which can be extended to multiple object tracking in crowd. They select for each individual in the scene a set of points and characterize them by computing 2D HOG descriptors, then they track these descriptors to construct temporal HOG descriptors.

Our approach uses statistical pre-defined models of scenarios to detect crowd events in video frames. The utilization of these pre-defined models allows us a more flexible and general way to model scenarios. We use object tracking to estimate crowd direction and speed, in lieu of using a holistic approach for its higher accuracy. Others approaches use also object tracking as in [12] (optical flow), however our approach is more robust because we are using HOG descriptors which better characterized the tracked points.

3. Crowd Tracking

This section describes the tracking process for crowd through the feature points computed for every object detected in a frame. We briefly describe the object detection process which does not belong to our contribution.

To perform object detection we use the technique proposed by *Nghiem et al.* [17] to calculate the difference between the current image and the reference one (background). The idea is to set up the moving regions by grouping foreground neighbouring pixels, where moving regions are classified into objects depending on their size (crowds, persons, groups, etc.).

Once the moving objects are detected in the scene using moving segmentation we track these objects by tracking the feature points

3.1 Feature Points

After obtaining the detected moving objects in the current frame, we compute for each of them a set of feature points to track. For this, we use FAST approach [2]. However, any other corner detector approach could be applied like the one proposed by *Shi et al.* in [18]. Our method consists in a descendant sort out of the detected feature points using corner strength information. Then, from these points (beginning from the most significant, i.e. the one with the biggest value of corner strength) a subset of feature points is chosen to ensure a minimum distance: between them. And also between all tracked points in the corresponding object. The minimum distance improves the feature point distribution for an object and prevents mixing tracked points.

3.2 2D HOG Descriptor

We build a HOG descriptor [3] for each detected feature point. To compute the descriptor we define around the feature point a block of 9 cells (3×3) where a cell is defined by a matrix of $p \times p$ pixels ($p \in \{3, 5\}$). Then, we compute the approximate absolute gradient magnitude (normalized) and gradient orientation for every pixel in the block using Sobel operator. Using gradient orientation we assign to each pixel from a cell one of the K orientation bins (by default $K = 9$). For each bin, we compute the sum of gradients of its pixel. Finally, we obtain for each cell inside a block a feature vector of K orientation bins. The 2D descriptor is then a vector for the whole block, concatenating the feature vectors of all its cells normalized by p.

3.3 Descriptor Tracking

The feature points detected in the previous frame are tracked in the current frame using the 2D HOG descriptors. In the current frame we calculate the mean over the trajectory, S_{GM}, of an object speed within a time window using all speed values from the feature points that belong to the same object. If the feature point is newly detected in the current frame we assume that $S_{GM} = S_{mean}$, where S_{mean} is the mean speed of the object at the current frame. To reduce the processing time we are using a searching window which is define based on a searching radius. For a given feature point F, the searching radius, R_s, is computed:

$$R_s = S_{GM} + \frac{1}{T} \times (S_{mean} - S_{GM}) \tag{1}$$

Where T is the number of frames where F was tracked. From equation (1), R_s is more accurate when F has a longer track.

The difference between two HOG descriptors, d^n and d^m, is defined by the equation:

$$E(d^n, d^m) = \sum_{i=1}^{9 \times K} MAX(v_i^n, v_i^m) \times (d_i^n - d_i^m)^2 \tag{2}$$

Where v^n and v^m correspond to the variances of the HOG descriptors of d^n and d^m, respectively, computed through out the time window.

Finally, we track F in the current frame comparing the difference (equation (2)) of the HOG descriptors between F and r ($\forall r$ a point inside the window of radius R_s). We choose the point r' in the current frame which better matches with the point F in the previous frame by computing the difference between their HOG descriptors.

We update the HOG descriptor of the tracked point by computing the equation below:

$$d_i^F = (1-\alpha)d_i^r + \alpha d_i^F, \qquad i = 1 \ldots 9 \times K \quad (3)$$

Where d_i^F is the mean HOG descriptor and d_i^r is the HOG descriptor of the point r in the current frame. α is a cooling parameter. In the same way, to update the variance of the mean descriptor bin in the current frame:

$$v_i^F = (1-\alpha) \times |d_i^r - d_i^F| + \alpha v_i^F, \quad i = 1 \ldots 9 \times K \quad (4)$$

4. Crowd Event Recognition

Crowd behavior can be characterized by regular motion patterns like direction, speed, etc. For this reason, the most robust and simple approach for crowd event recognition is to use pre-defined models of crowd events. In this section, we explain the crowd motion information computed to define and recognize the different crowd events used in this study.

Our approach consists in modeling crowd events through the information obtained with the tracking of the feature points. We rely on those motion vectors of feature points computed over multiple frames. For us a *vector* is a collection of several elements which are the mean HOG descriptor, the start and end point of the trajectory of the tracked feature point, together with start and end time. The computed attributes (information) related to motion vectors are *direction*, *speed*, and *crowd density*.

Direction is the property that identifies the direction of the trajectory of feature points (called vectors). We divide the Cartesian plane into 8 parts where each part is a direction between the angles [α, α + 45] and $\alpha \in \{0, 45, 90, 135, 180, 225, 270, 315\}$, see Figure 1. The angle of the vector is computed between the axis X (where $x = 0$ is the starting direction of the vector) and the vector, this measure decides in which of the 8 directions is classified the vector. After this, we calculate the principal crowd directions considering the density percentage of feature points in each direction. If this percentage is bigger than a threshold t we assume there is a crowd in that direction.

The speed is directly related to the length of the vectors. For each frame we calculate the speed of every vector considering its length and the number of tracking frames of the feature point associate to the vector. We obtain the crowd average speed using the speed of all the vectors in the frame.

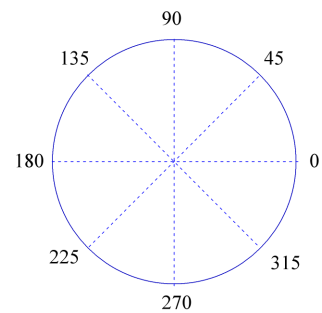

Figure 1: Directions in the Cartesian Plane

For crowd density we build a grid over the image and then we compute the density of feature points in each grid-cell. This information will help us to recognize the crowd events. 6 crowd events are modeled, which are walking, running, evacuation, local dispersion, crowd formation and crowd splitting. The models defined for this study are described below:

- **Walking**: corresponds to a significant number of individuals moving at a low speed. We compute the mean speed, measured as pixels per frame, considering all vectors in a frame. We set up the threshold t_1, and when the mean speed is under this threshold we recognize a *crowd walking* event.
- **Running**: corresponds to a significant number of individuals moving at a high speed. We compute the mean speed, measured as pixels per frame, considering all vectors in a frame. We use the same threshold t_1, but when the mean speed is over t_1 we recognize a *crowd running* event.
- **Evacuation**: corresponds to a rapid dispersion of the crowd in different directions. We use the attributes direction and crowd density to recognize this event. When there are more than 4 principal directions, when the minimum distance between the principal directions is over a threshold t_2 (euclidean distance between the grid-cells containing the feature points related to principal directions), and if the addition of the crowd density per principal direction is over a threshold t_3, this event is recognized.
- **Crowd Formation**: corresponds to the merge of several individuals, where the individuals approach from different directions. Crowd density and the distance between the principal directions are used to model the current event. We set up the thresholds t_4 for the distance between the principal directions, and t_5 for the crowd density in the respective grid-cells. When the minimum distance is under t_4 and the crowd density is over t_5, a *crowd formation* event is recognized.

- **Crowd Splitting**: corresponds to a cohesive crowd of individuals which splits into two or more flows. The crowd density and the distance between the principal directions are used to model the current event. We set up the thresholds t_6 for the distance between the main directions, and t_7 for the crowd density in the respective grid-cells. When the maximum distance is over t_6 and the crowd density is under t_7, a *crowd splitting* event is recognized.

- **Local Dispersion**: corresponds to localized movement of people within a crowd away from a given threat. This event is very similar to crowd formation/splitting because this model uses the same attributes, plus another one: the speed. Nevertheless the thresholds (also used for crowd formation/splitting) are different. Moreover, the threshold for the distance between the grid-cells has to be over a threshold t_8 and the crowd density has to be distributed between the grid-cells with more than 1 principal directions. The mean speed has to be under a threshold t_9.

Table 1: Measures to evaluate the approach

Crowd Event	TP	FP	SN
Crowd Splitting	844	222	0.79
Crowd Formation	637	430	0.60
Walking	976	90	0.92
Running	982	85	0.92
Evacuation	1035	31	0.97
Local Dispersion	778	230	0.77

Table 2: Frame number for each Video Clip

Name VC	First Frame	Last Frame
14:16-A	0	107
14:16-B	108	222
14:27-A	0	184
14:27-B	185	333
14:31	0	130
14:33-A	0	310
14:33-B	311	377

5. Experimental Results

To validate our approach we have tested the PETS Dataset **S3**, High Level, which contains four sequences respectively with timestamps $14:16$, $14:27$, $14:31$ and $14:33$. For each sequence we use the videos recorded by *camera 1* (View 1), and we consider that there are two video clips inside the sequences $14:16$, $14:27$ and $14:33$ and one video clip for the sequence $14:31$. A video clip is about 130 frames long. The videos depict the 6 crowd scenarios described in the previous section. The crowd scenarios are acted by about 40 people from Reading University Campus. All the experiments have been performed on one view and our plan is to complete the experiments on the other views.

The thresholds used in the event models have been set up experimentally. We are currently designing a learning process to compute and optimize the thresholds.

Table 1 presents some measures to evaluate our approach: true positives (TP), false positives (FP) and sensitivity (SN). We consider TP as the crowd event that matched with the ground trouth for each frame, FP as the not matched crowd event recognized for each frame, and SN is defined as $TP/(TP+FN)$. Since the ground truth is not established for the S3 High Level, we have built the ground truth manually.

Table 2 contains the frame number of the 7 videos clips.

Table 3 shows the significant time intervals where the pre-defined events were recognized for the 7 videos clips. The columns are the different videos. There are 6 rows which represent the crowd scenarios in our study. Each element of the table contains the frames where the event is recognized in the corresponding video clip. The video clips named time_stamp-B are the continuation of the video sequence time_stamp, i.e. if the last frame of time_stamp-A is 104 the first frame of time_stamp-B is 105. Inside the brackets two time intervals are separated by ";". Significant time interval is when the size is bigger than 9 frames. False positives of crowd event can be detected as significant time intervals.

Figure 2 shows some illustrations of the results of our approach. The black lines are the trajectories of the tracked feature points depicting their direction and length.

6. Conclusion

In this paper we have presented a novel approach for recognizing crowd events. The contributions are the combination of local and global analysis. The local analysis is achieved by tracking HOG descriptors and the global analysis is obtained by statistical analysis of the HOG motion patterns. Also, the use of HOG descriptors for tracking enables a high accuracy in crowd event recognition and a better characterization of feature points. The approach has successfully validated on PETS dataset. There are still some errors in the recognized events. These errors are mainly do to the set up the thresholds at the level of scenario models. For future work we plan to improve the threshold computation by automating the construction of scenario models. We are also currently computing the HOG motion vectors in 3D for the approach to be independence from the scene. The scenario

Table 3: Time Intervals for Crowd Events Recognized

Crowd Event	**14:16-A**	**14:16-B**	**14:27-A**	**14:27-B**	**14:31**	**14:33-A**	**14:33-B**
Crowd Formation	[21,102]	[]	[25:45 ; 89:167]	[201,213 ; 267,313]	[9,97]	[69,157 ; 253,310]	[332,341]
Crowd Splitting	[]	[]	[]	[]	[98,130]	[12,52; 158,251]	[363,377]
Walking	[1,37]	[109,174 ; 201,222]	[1:184]	[186,333]	[1,130]	[1,310]	[313,341 ; 348,377]
Running	[38,107]	[175,200]	[]	[]	[]	[]	[342,347]
Evacuation	[]	[]	[]	[]	[]	[]	[342,347]
Local Dispersion	[]	[]	[47:55]	[]	[98,130]	[158,251]	[]

models (besides the thresholds) are easy to model by users and can be extended to other crowd scenarios. Definition of a language for modeling these scenarios can also enhance the flexibility of the approach to pre-define the scenarios.

References

[1] B. Zhan, P. Remangino, D.N. Monekosso, and S.Velastin, "The Analysis of Crowd Dynamics: From Observations to Modeling," *Computational Intelligence*, Springer Berlin Heidelberg V.1, pp. 441-472, 2009.

[2] E. Rosten, and T. Drummond, "Machine Learning for High-Speed Corner Detection," *In European Conference on Computer Vision*, Vol. 1, pp. 430-443, Austria: Springer, May 7-13, 2006.

[3] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *In International Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886-893, San Diego, CA, USA: IEEE Computer Society Press, Jun 20-25, 2005.

[4] M. Hu, S. Ali, and M. Shah, "Detecting Global Motion Patterns in Complex Videos," *In ICPR'08: International Conference on Pattern Recognition*, 2008.

[5] Y. Benabbas, N. Ihaddadene, and C. Djeraba, "Global Analysis of Motion Vectors for Event Detection in Crowd Scenes," *In Proceedings 11th IEEE: International Workshop on PETS 2009*, pp. 109-116, Miami, USA, Jun 25, 2009.

[6] A. Marana, S. Velastin, L. Costa, and R. Lotufo, "Estimation of Crowd Density Using Image Processing," *IEEE Colloquium Image Processing for Security Applications (Digest No.:1997/074*,pp. 11/1-11/8, 1997.

[7] S. Lin, J. Chen, and H. Chao, "Estimation of Number of People in Crowded Scenes Using Perspective Transformation," *ISystems, Man and Cybernetics, Part A, IEEE Transactions*, 31(6):645654, 2001.

[8] R. Ma, L. Li, W. Huang, and Q. Tian, "On Pixel Count Based Crowd Density Estimation for Visual Surveillance ," *Cybernetics and Intelligent Systems, 2004 IEEE Conference*,vol. 1:170173, 2004.

[9] H. Rahmalan, M. S. Nixon, and J. N. Carter, "On Crowd Density Estimation for Surveillance ," *In International Conference on Crime Detection and Prevention*, 2006.

[10] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *International Journal of Computer Vision*, 63(2): 153-161, 2005.

[11] X. Huang, and T. Sim, "Stereo-Based Human Head Detection from Crowd Scenes," *Proc. International Conference on Image Processing*, 1353-1356, 2004.

[12] N. Ihaddadene, and C. Djeraba, "Real-Time Crowd Motion Analysis," *ICPR International Conference on Pattern Recognition*,2008.

[13] R. Mehran, A. Oyama, and M. Shah, "Abnormal Crowd Behavior Detection Using Social Force Model," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,2009.

[14] R. Cucchiara, C. Grana, G. Tardini and R. Vezzani, "Probabilistic People Tracking for Occlusion Handling," *ICPR 17th International Conference on Pattern Recognition*, Vol. 1, pp. 132-135, 2004.

[15] B. Lucas, and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *International Joint Conference on Artificial Intelligence*, 674-679, Vancouver, Canada, 1981.

[16] M. Kaaniche, and F. Bremond, "Tracking HOG Descriptors for Gesture Recognition," *In AVSS'09: 6th IEEE International Conference on Advanced Video and Signal Based Surveillance* , Genoa, Italy, 2009.

[17] A. Nghiem, F. Bremond, and M. Thonnat, "Shadow Removal in Indoor Scenes," *In AVSS'08: 5th IEEE International Conference on Advanced Video and Signal Based Surveillance* , Santa Fe, USA, September 1-3, 2008.

[18] J. Shi, and C. Tomasi, "Good Features to Track ," *In International Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA: Springer, June, 1994, pp. 593-600.

Figure 2: The first row presents the original frames and the second row the output of our approach